Electron Cooling BPM User's Guide

Table of Contents

Table of Figures

Table of Tables

Introduction

Electron cooling is one component of the Run IIb upgrades targeted at increasing integrated luminosity at Fermilab. Briefly, the project involves the use of an electron beam to reduce (cool) the longitudinal and transverse emittances of anti-protons stored in the Recycler. The electron cooling project complements the present stochastic cooling in the Recycler and should permit "...faster stacking and larger stacks, and ultimately to re-cool (recycle) anti-protons, which remain at the end of Tevatron stores."[1] The electron beam will propagate co-linearly with the anti-proton beam in a short section of the Recycler at MI-30. Scattering between the particles and the consequent friction results in thermal equilibration between the particles and thus cooling. A more complete discussion of the mechanism of electron cooling and its implementation at Fermilab can be found elsewhere.[2]

Electron cooling beam position monitoring

In order to support the goals of the electron-cooling project, accurate beam position measurement is essential to maintaining the co-propagating beams of electrons and antiprotons. We are interested in measuring the relative positions of electron and anti-proton beams to within a ± 100 μm. In order to separate the signals from each beam, the electron beam will be modulated at a frequency of 32 kHz. The digital signal receiver processing electronics (DSR) has the ability to discriminate between different signal frequencies by down-converting an input frequency to base band using a numerically controlled oscillator. The circulating anti-proton beam will be detected at the beam revolution frequency and the electron beam will be detected at 32 kHz. Subsequent processing of these signals will produce beam position measurements exposed to the accelerator community as Acnet read-only variables.

Operational vs devevlopment systems

The electron cooling system was installed at MI-31 during the Fall 2004 shutdown. The system was commissioned for operations in the spring of 2005 and the first demonstrated electron cooling was shown in the summer. The operational BPM system located at MI-31, consists of two VXI crates and is mirrored by a development system, stored in the LLRF VXI laboratory. We have installed electron cooling BPM hardware and software at both locations. The two systems have unique node names, `ecbpm` – operational, `ecbpma` – operational (second crate) and `ecbpmd` – development, with distinct Acnet database trunk and node values. The exact physical location of the nodes may change over the longer term however their semantic meaning will remain fixed. The nodes, `ecbpm` and `ecbpma`, will always refer to the operational system. Our LLRF convention for Acnet devices referring to `ecbpm` specific devices follows the form **R:V***XXXXX* , those referring to `ecbpma` specific devices as **R:A***XXXXX* and those referring to `ecbpmd` will follow the form **Z:E***XXXXX*, where *XXXXX* will vary depending on the specific device. Similar devices on both systems will have identical *XXXXX* suffixes and differ only in the prefix. I will use the () notation to denote the device prefix where we do not need to be specific about the system owning the device.

This general device naming convention is violated in some specific places. In order to support the electron cooling group and a “problem domain” understanding of the system we have adopted their conventions for BPM names and have used suffixes to distinguish between variables. The following mapping applies

Table 1: BPM Channel mappings and names for the first VXI crate (ecbpm).

Device Name	DSR Slot (Port)	Channel Pair	Description
R:BXA01*	1 (Top)	A	Acceleration BPM Horizontal
R:BYA01*	1 (Top)	B	Acceleration BPM Vertical
R:BXS02*	1 (Bottom)	C	Supply BPM 2 Horizontal
R:BYS02*	1 (Bottom)	D	Supply BPM 2 Vertical
R:BXS03*	2 (Top)	A	Supply BPM 3 Horizontal
R:BYS03*	2 (Top)	B	Supply BPM 3 Vertical
R:BXS04*	2 (Bottom)	C	Supply BPM 4 Horizontal
R:BYS04*	2 (Bottom)	D	Supply BPM 4 Vertical
R:BXS05*	3 (Top)	A	Supply BPM 5 Horizontal
R:BYS05*	3 (Top)	B	Supply BPM 5 Vertical
R:BXB01*	3 (Bottom)	C	BPM Prior to CS Horizontal
R:BYB01*	3 (Bottom)	D	BPM Prior to CS Vertical
R:BXC00*	4 (Top)	A	Cooling BPM 00 Horizontal
R:BYC00*	4 (Top)	B	Cooling BPM 00 Vertical
R:BXC10*	4 (Bottom)	C	Cooling BPM 10 Horizontal
R:BYC10*	4 (Bottom)	D	Cooling BPM 10 Vertical
R:BXC20*	5 (Top)	A	Cooling BPM 20 Horizontal
R:BYC20*	5 (Top)	B	Cooling BPM 20 Vertical
R:BXC30*	5 (Bottom)	C	Cooling BPM 30 Horizontal
R:BYC30*	5 (Bottom)	D	Cooling BPM 30 Vertical
R:BXC40*	6 (Top)	A	Cooling BPM 40 Horizontal
R:BYC40*	6 (Top)	B	Cooling BPM 40 Vertical
R:BXC50*	6 (Bottom)	C	Cooling BPM 50 Horizontal
R:BYC50*	6 (Bottom)	D	Cooling BPM 50 Vertical
R:BXC60*	7 (Top)	A	Cooling BPM 60 Horizontal
R:BYC60*	7 (Top)	B	Cooling BPM 60 Vertical
R:BXR70*	7 (Bottom)	C	Cooling BPM 70 Horizontal
R:BYR70*	7 (Bottom)	D	Cooling BPM 70 Vertical
R:BXR80*	8 (Top)	A	Cooling BPM 80 Horizontal
R:BYR80*	8 (Top)	B	Cooling BPM 80 Vertical
R:BXC90*	8 (Bottom)	C	Cooling BPM 90 Horizontal
R:BYC90*	8 (Bottom)	D	Cooling BPM 90 Vertical
R:BXQ01*	9 (Top)	A	BPM following CS Horizontal
R:BYQ01*	9 (Top)	B	BPM following CS Vertical
R:BXR01*	9 (Bottom)	C	Return BPM 1 Horizontal
R:BYR01*	9 (Bottom)	D	Return BPM 1 Vertical

R:BXR02*	10 (Top)	A	Return BPM 2 Horizontal
R:BYR02*	10 (Top)	B	Return BPM 2 Vertical
R:BXR03*	10 (Bottom)	C	Return BPM 3 Horizontal
R:BYR03*	10 (Bottom)	D	Return BPM 3 Vertical
R:BXR04*	11 (Top)	A	Return BPM 4 Horizontal
R:BYR04*	11 (Top)	B	Return BPM 4 Vertical
R:BXR05*	11 (Bottom)	C	Return BPM 5 Horizontal
R:BYR05*	11 (Bottom)	D	Return BPM 5 Vertical

The asterisk is used to denote a prefix attached to the device name to indicate a given variable. The suffixes in use are

Suffix	Description
F	Fast (High BW) BPM position data
S	Slow (Low BW) BPM position data
I	Intensity signal for a BPM channel

This mapping implies that the correct BPM cable must be attached to the correct BPM input for the Acnet database device entries to give "meaningful" data.

Table 2: BPM Channel mappings and names for the second VXI crate (ecbpma).

Device Name	DSR Slot (Port)	Channel Pair	Description
R:BXR06*	1 (Top)	A	Return BPM 6 Horizontal
R:BXR06*	1 (Top)	B	Return BPM 6 Vertical
R:BXT02*	1 (Bottom)	C	Transfer BPM 2 Horizontal
R:BYT02*	1 (Bottom)	D	Transfer BPM 2 Vertical
R:BXT03*	2 (Top)	A	Transfer BPM 3 Horizontal
R:BYT03*	2 (Top)	B	Transfer BPM 3 Vertical
R:BXT04*	2 (Bottom)	C	Transfer BPM 4 Horizontal
R:BYT04*	2 (Bottom)	D	Transfer BPM 4 Vertical
R:BXT05*	3 (Top)	A	Transfer BPM 5 Horizontal
R:BYT05*	3 (Top)	B	Transfer BPM 5 Vertical
R:BXD07*	3 (Bottom)	C	Deceleration BPM 7 Horizontal
R:BYD07*	3 (Bottom)	D	Deceleration BPM 7 Vertical
R:BXD05*	4 (Top)	A	Deceleration BPM 5 Horizontal
R:BYD05*	4 (Top)	B	Deceleration BPM 5 Vertical
Unused			
Unused			
R:BXA07*	5 (Top)	A	Acceleration BPM 7 Horizontal
R:BYA07*	5 (Top)	B	Acceleration BPM 7 Vertical
R:BXD08*	5 (Bottom)	C	Deceleration BPM 8 Horizontal
R:BYD08*	5 (Bottom)	D	Deceleration BPM 8 Vertical
	6 (Top)	A	Damper BPM (C00) Horizontal
	6 (Top)	B	Damper BPM (C00) Vertical

	6 (Bottom)	C	Damper BPM (C10) Horizontal
	6 (Bottom)	D	Damper BPM (C10) Vertical

Console Application

E50 User Interface

The ECOOL BPM system has been constructed to provide information on real-time beam positions for the electron beam, generated by the Pelletron, and circulating beam, antiproton or proton, in Recycler. Configuration of the system for a given measurement can be done using the E50 BPM/BLM Plots primary application.

As discussed previously, there are two ECOOL BPM systems that we distinguish as "development" and "operational". From E50, pull-down menu selects the system. The "operational" system is E-COOLRR and the development system is E-COOLDV. After the system is selected a new window will display with a menu bar shown at the top and 4 sub-windows as shown in Fig.1 for E- COOLRR. There are five entries on the menu bar: DATA DISPLAY, PLOT SETUP, ARCHIVE, TIMING and UTILITY. Each menu item has a corresponding sub-window shown under the menu bar except the UTILITY item. At the bottom of the window, there is a message window. A view of the E50 main page for the E-COOL BPM system is shown in Figure 1.

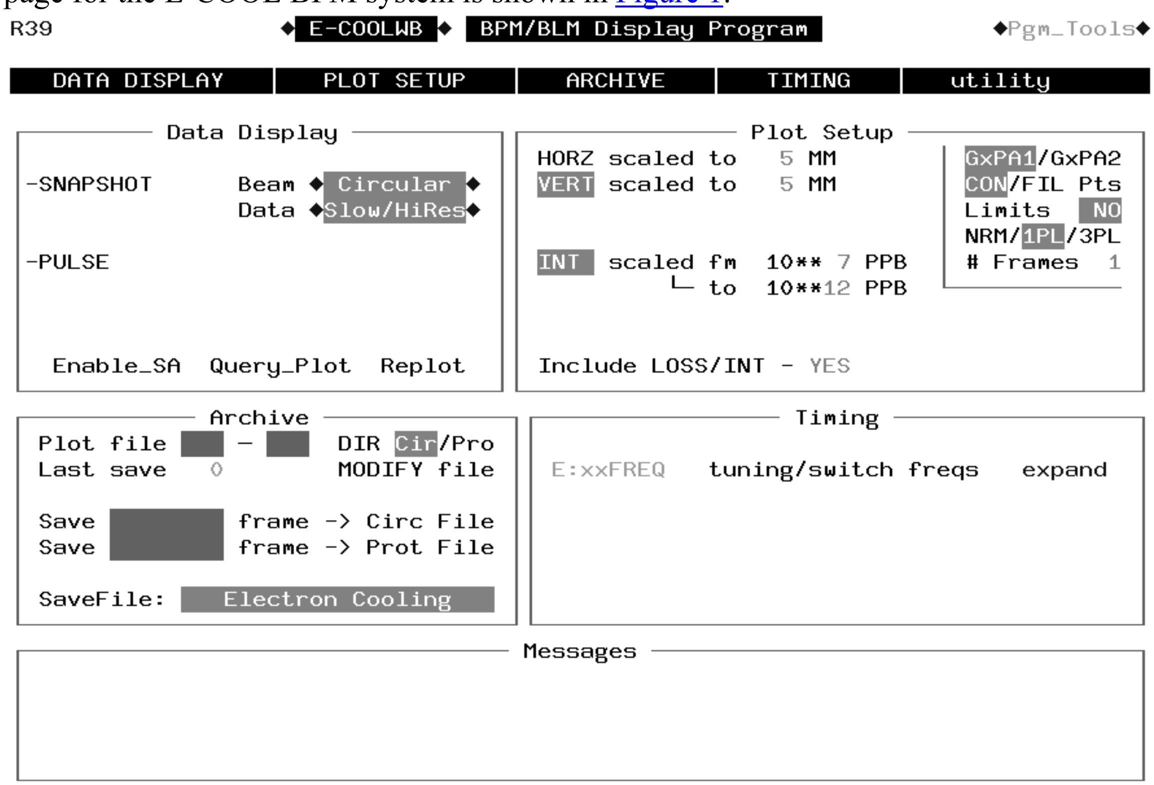

Figure 1: E50 PA window for Electron Cooling operational system, the window appears as it would when it has just be selected from the system menu. Colors are inverted from normal PA for clarity.

The basic function of each sub-window is similar to other E50 PA pages. A brief description with E-COOL BPM specific information is included here.

- DATA DISPLAY – enables user to select which beam (electron or circulating) to probe, the type of filtering to use for the position data to display and the operational mode to use in the display. This will be described further in the following section.
- PLOT SETUP – enables the user to select the position data orientation, horizontal or vertical, to display, the plot destination and scaling for position and intensity.
- ARCHIVE – enables the user to store the generated plot in a file
- TIMING – enables the user to change the frequency used to detect either the electron or circulating beam.
- UTILITY – enables the user to perform E-COOL specific configuration. This window is initially hidden and displayed only when the UTILITY menu is selected; selecting this sub-window hides all other sub-windows.

Data Display – Operational Modes

There are four beam position measurement modes:

1. Electron beam position measurement mode
2. Circulating beam position measurement mode
3. Switched beam position measurement mode
4. Electron pulsed beam position measurement mode.

The end user can invoke any of the above modes from DATA DISPLAY sub-window. Within this sub-window there are two pop-up menus: BEAM and DATA. The BEAM pop-up menu selects the beam type to measure electron, circulating or both, which corresponds to the switch measurement mode. Electron pulsed beam mode will be discussed later. The DATA pop-up menu options, SLOW or FAST, select the filtering to apply to the position data. The SLOW position data is generated from a narrow band filter (BW < 5 Hz), and is the averaged output of the FAST positions; this reduces the noise in the position data, relative to the FAST position data, at the expense of a longer response time. The end user can change the tuning frequency for the corresponding measurement ()CBPME or ()CBPMP from the TIMING sub-window. In general, the tuning frequency is not necessarily the same as the modulation frequency. The end user can move the tuning frequency off the modulation frequency to avoid saturation in the DSR modules. A common indicator of saturation is high beam intensity and unstable fluctuation of beam position measurements. A block diagram showing the CW signal processing is shown in

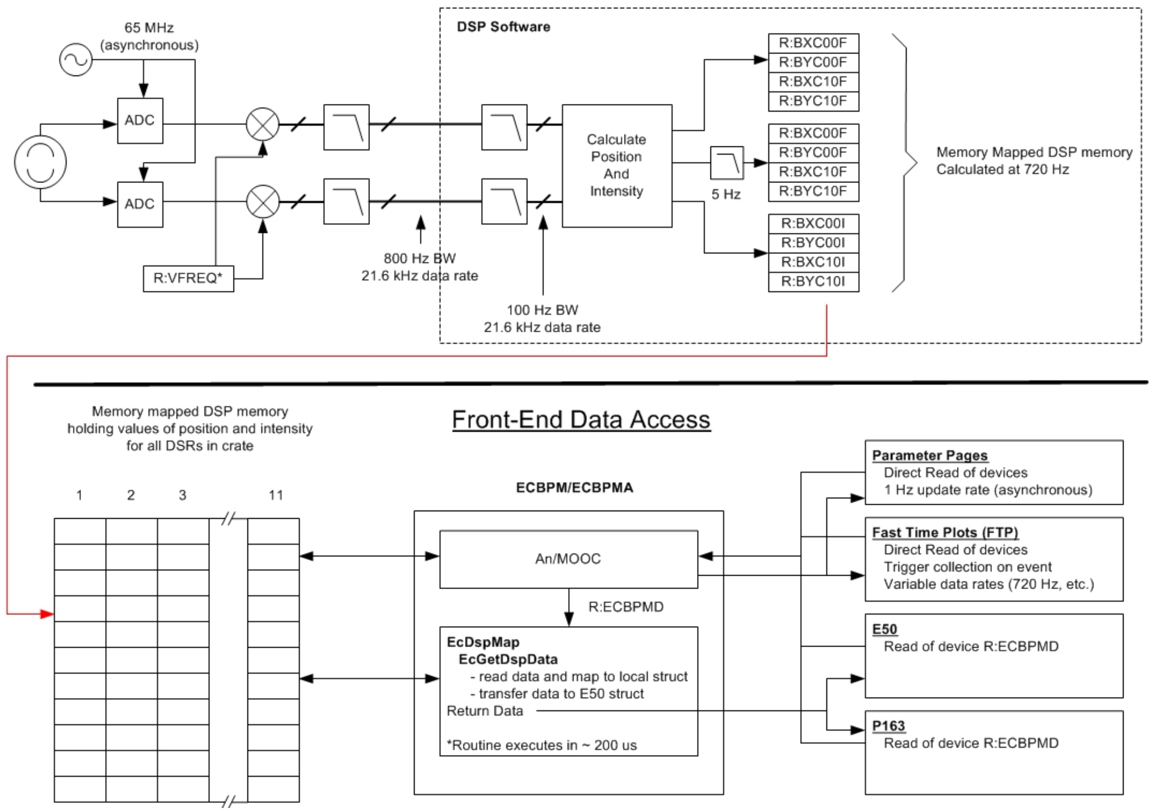

Figure 2: CW processing for ECBPM/A system.

While in the electron or circular measurement mode, the ECOOL BPM system provides horizontal and vertical positions for either electron or circular beam. The timing of the measurement is controlled by the TCLK 0xEB event. This event can be tied to other TCLK events and delayed from these events by a specified amount using the device R:ECDLY, thus allowing the end-user to accurately time data collection relative to other accelerator events. Once the 0xEB event is received, the front-end collects data from the DSPs for all BPMs and maps that to an intermediate data structure accessed by the E50 (or P163) application. In the switch mode, ECOOL BPM system alternates between measuring beam positions for both beams. The frequency with which the measurement alternates between electron and circular beams can be changed with the acnet device R:AMFREQ in the TIMING sub-window.

An additional Acnet device, ()xAERx, displays the calculated separation between two beams. Suppose we have measured horizontal and vertical components *X*e and *Y*e for the electron beam and *X*c and *Y*c for the circular one. Then, the separation between two beams is calculated as

$$\sqrt{(Xe - Xc)^2 + (Ye - Yc)^2}$$

Instead to look into four components in electron and circular measurement modes, switch measurement mode can tell if electron and circular beam is well aligned with a single variable. Especially, the fast time plot of this Acnet devices over the time can be very helpful in monitoring the alignment between electron and circular beams. Ideally, above FTP variable will a very small number if both beam is well aligned.

Once the measurement type, BEAM and DATA, has been set, a click of SNAPSHOT in the DISPLAY DATA window invokes the corresponding measurement mode. The selected mode persists until the end user selects another measurement mode. The message window will display the progress of the operation with messages such as, “ Start Electron measurement mode …”, and “ Electron measurement, success “ if running electron measurement mode is successful. Errors will also display in the message window. Messages from different sources appear in the message window with different colors; messages from the ECOOLBPM system will appear in yellow. The instantaneous beam positions measured for all BPMs in the system will be shown in a new pop-up window. Continuous beam positions are also available as Acnet devices, which can be accessed with fast time plots. The ranges of operational BPM intensities large enough to give reliable positions information are shown in Table 3. Position values are calculated and displayed for intensities above 10, below 10 positions are clamped to 0. The noise in calculated position is noticeably higher for intensities lower 150 in DC beam.

Table 3: Operational BPM intensity ranges.

Beam Type	BPM Operating Mode	Pre-amp Gain Setting	Recommended Intensity Range
DC	Electron or Circular	High	150–25000
DC		Low	150–25000
Pulsed	Pulsed	High	10–300
Pulsed		Low	10–300

Electron Beam Pulsed Mode

The electron pulse measurement mode can be invoked by clicking or interrupting on the –PULSE device. One can also turn the pulse measurement mode on or off from Acnet device ()PULSE, depending on which system is running. The status of this acnet device also tells if the system is in pulsed measurement mode. There is time delay t_{inherent} between the electron pulse Tclock event trigger and data acquisition due to the digital signal processing. This Acnet device also provides a mechanism for fine tuning by changing its setting value, t_{acnet}. In general, we should observe the higher beam intensity if t_{acnet} well matches t_{inherent}.

In electron pulse measurement mode, the data acquisition is trigged by the electron pulse Tclock event 0xEB. It is important to check the status of the ECOOL Variable Pulse Delay Acnet device –R:ECDLY should be on. When R:ECDLY is off, a red star will be shown, then, the end user should pop-up the D/A menu, and switch to Digital Control, then double click on the ON, it will make the electron pulse event available..

Calibration Procedure

A click on the UTILITY menu bar item pops up the following window

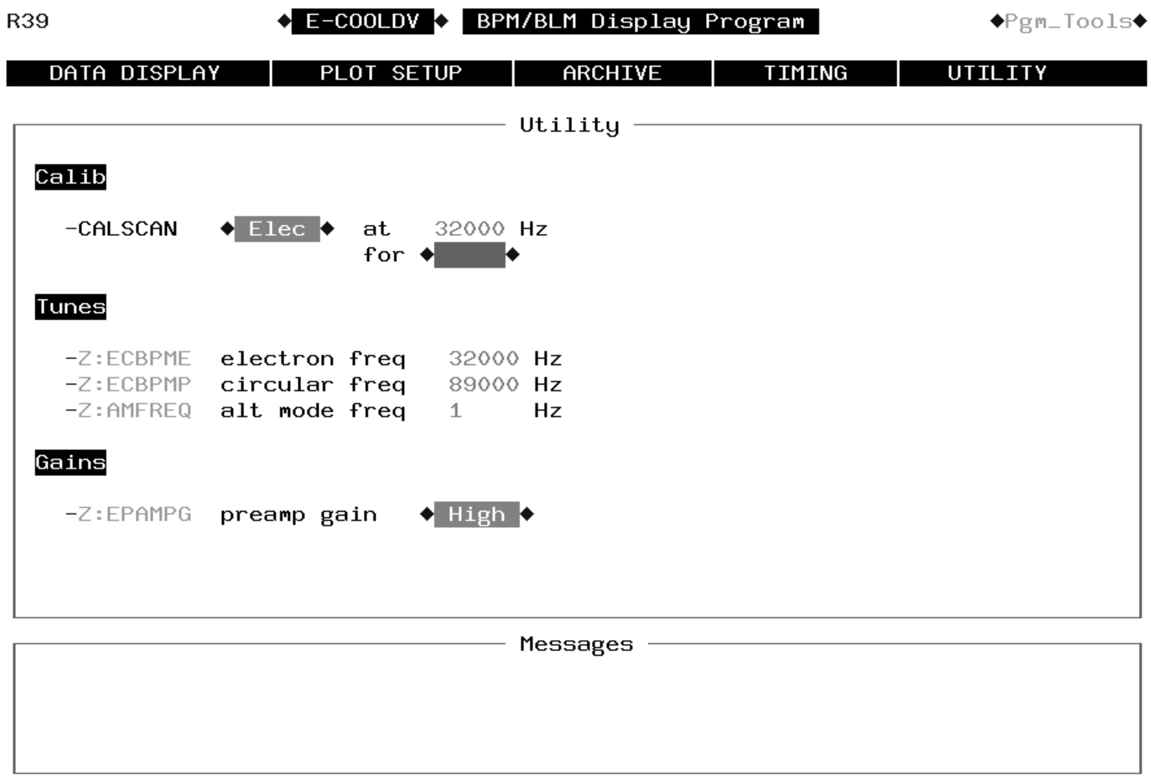

Figure 3: E50 PA UTILITY menu sub-window for Electron Cooling development system, an identical sub-window exists for the operational system. Colors are inverted from normal PA for clarity.

This window allows the end user to perform a system calibration. The key steps arR:

1. Set appropriate frequency and amplitude for the signal generator in the calibration system. The frequency and amplitude are settable using a LabView application (Rabbit UI) residing on the OTR computer in the MI-31 control room. Table 4 details suitable amplitude and frequency levels for these devices. The frequency must be set first and then the amplitude. Ensure system is not saturated. The amplitude should be set to a value such that the reading from the Acnet of beam intensity close matches to that in the normal working situation. To

Table 4: Calibration signal device settings for BPM calibration.

BPM Operating Mode	Gain Setting ( )PAMPG	Frequency (Hz)	Amplitude (Vpp)
Electron	High Low	32050	0.4 4.0
Circular	High Low	89050	0.4 4.0

2. Select the calibration beam type from the pop-up menu, either electron or circular. Then. Set the calibration frequency. The calibration frequency should be the same

as that set at the signal generator of the calibration system. Also select the DSR module for which one wishes to calibrate.

3. Select the pre-amplifier gain, either high or low in the same window.
4. Then, click on –CALSCAN, the system will start the calibration. Meanwhile, watch the message window. In usual, you will see messages “Calibration is initiated …” and “Calibration is success” if the calibration goes smoothly.
5. If calibration is successful, the calibration factors will be updated automatically. Meanwhile, the new calibration factors will also be saved into the fecode-bd file server and Acnet device database. The saved calibration factors will be reloaded to the DSR DSP memory automatically when the system is rebooted.
6. One can inspect the calibration factors from the hidden UTILITY page; directions are given below.
7. Turn off the calibration signal by setting the amplitude to 0.01 or less, 0 is fine.

One can also visually inspect Acnet devices of the beam positions to see if it the calibration was performed correctly. Calibration values of exactly 1 are generated if the BPM intensity for that channel is below 50. This can indicate problems with either the pre-amp or the DSR card, however first check all cable connections for the suspect channel to ensure that the calibration signal is reaching the DSR card. The calibration is performed in DSR DSP. First, DSR is put into a calibration state, in which a loop of 100 is applied to calculate the average beam position, then, compare the calculated average beam position to the predicted one. In this way, the calibration factor can be acquired. After that, the DSR is put back to the normal working state. The system applies the calculated calibration factors instantly to correct the beam position calculation soon after the calibration process is completed. Therefore, it is recommended to check Acnet devices such as position readbacks to see if their readings are equal to the predicted beam position.

After the calibration is done, ensure that the calibration signal is off by settting the amplitude of the calibration signal generator, R:BPCAL to zero. This is done to minimize the interference of the calibration system to normal beam position measurement because the calibration system and BPM detector share the same input to the DSR system in the current design of the calibration system.

At this moment, there is no additional calibration for the electron pulse mode. The calibration factor for the circular beam applies equally to the electron pulse measurement mode.

Viewing the Hidden Page

To facilitate BPM system expert settings through E50, a hidden sub-window has been included. This page allows system experts to view gain and calibration factors, set gain parameters for DSR hardware filters and perform frequency scans. The hidden page is accessed from the UTILITY sub-window by interrupting (clicking) in a specific location in upper right corner the UTILITY sub-window, topmost empty row within the UTILITY frame. This location is 3 “spaces” from the right margin and will change to a diamond (♦) when the cursor is over it. When displayed the hidden page has the general appearance shown in Figure 3.

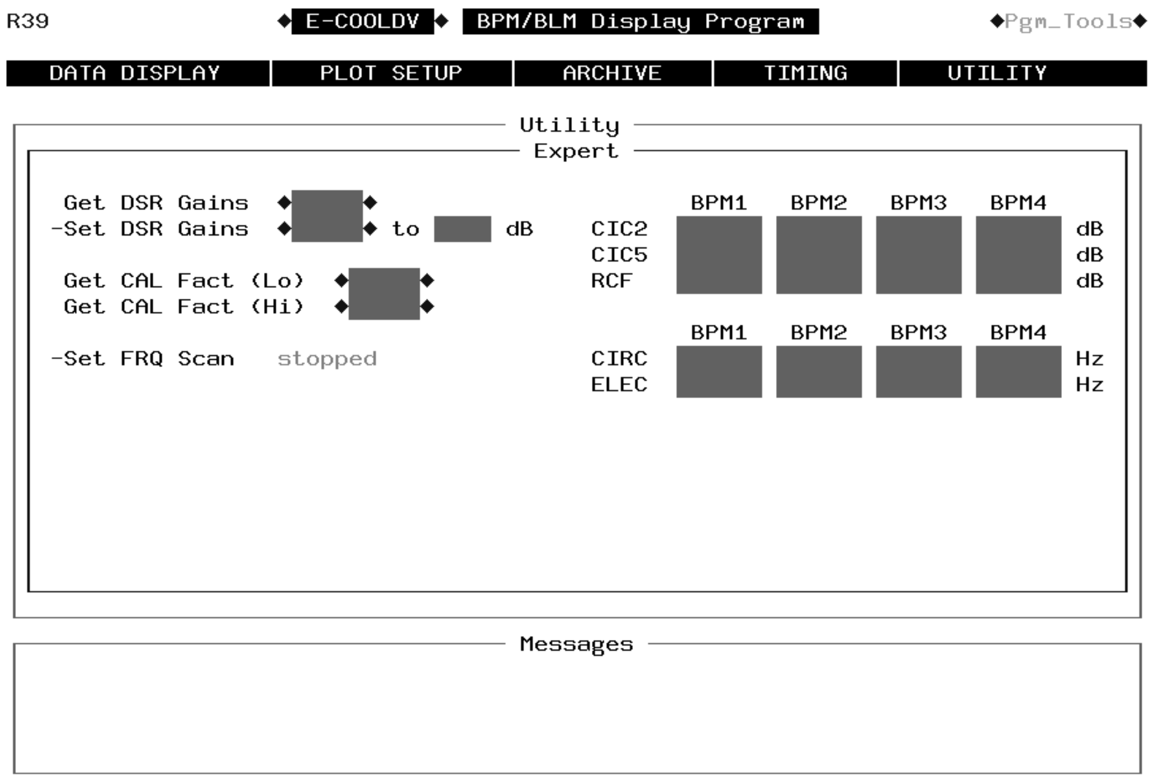

Figure 4: E50 PA UTILITY menu hidden sub-window for Electron Cooling development system, an identical sub-window exists for the operational system. Colors are inverted from normal PA for clarity.

Obtaining Information

The hidden page can be used to read the values of calibration constants or gain settings used by the DSR hardware filters.

To inspect the current setting for various filters for a given DSR module, select the DSR module of interest from the pop-up menu following the "Get DSR Gains", then, the current setting value of the gains for the filter CIC2, CIC5 and RCF for the four BPM channels of that DSR module will be shown in the 3 X 4 matrix to the right. The BPM channels are number from 1 to 4 on the hidden page (see Figure 4). BPM1 corresponds to the first pair of channels in the DSR module, BPM2 the second pair and so forth. Thus BPM1 and BPM2 refer to the BPM, generally horizontal and vertical inputs, on the top 15-pin D front-panel connector of a given E-COOL DSR module. BPM3 and BPM4 are the same inputs for a different BPM connected to the bottom 15-pin D front-panel connector.

To inspect the current calibration factors, determine the operational gain setting of the pre-amplifier, either Get CAL Fact (Low) or Get CAL Fact (High), then, select the DSR module of interest. The current values will be displayed in the 2 X 4 matrix to the right, below the gain reading matrix. The calibration values for both circulating beam and electron beam will be displayed here, the BPM notation is the same as above.

Set Gain for Various Filters in DSR Modules

The performance of the system is critically dependent on the setting of the gain in a series of filters in DSR modules. The system will become saturated if the gain is set too high. However, if the gain is set too low, the measurements will become much more noisy. In general, the gain should be set as high as possible as long as the system does not become saturated. To be able to set the gain appropriately is a basic skill to the end user, but not an easy one.

To set the gain for a filter, CIC2, CIC5 or RCF for all DSR modules existed in the system, select the type of filter and its gain in DB from two pop-up menu after "-Set DSR Gain", then, double click "-Set DSR Gains". The new gain setting value will be shown on the display block of the current gain setting after the setting of the gain is done in the system. The gain newly set will be saved to `fecode` file server and Acnet database automatically. It will be reloaded to the DSR DSP memory automatically when the system is rebooted next time.

For the sake of convenience, the current design of the gain setting from this window is to set the gain to all DSR modules in the system with the same value for the specified filter type. However, it is possible to set the filter gain for each individual DSR module and channel differently by using the Acnet devices ()DSRID, ()CHANX, ()CIC2G, ()CIC5G and ()RCFG.

A new feature for automatic gain control(AGC) is under development. An Acnet device ()AGCON has been created, and will be integrated in E50 on the same window below filer gain setting block. The Acnet device ()AGCON allows the end user to turn AGC ON or OFF. It will also show the current status of AGC is ON or OFF. When the AGC status is OFF, the end user can set the filter gain manually as discussed before. When the AGC is ON, the end user will see the updating of the gains on various filers while the intensity of the beam is changing. How often the gains are to be updated can also be set from the same Acnet device. The mechanism of AGC is based on the application of Tclock event 0x02 as timing system and current, such as R:IBEAM or R:IBEAM as reference, then empirically adjust the gains for various filters and pre-amplifiers periodically.

Frequency Scanning

The feature of the frequency scanning allows the end user to visually analyze the signal energy distribution over the frequency range of interest.

1. Set up a fast time plot of intensity ()*INTA (or other appropriate intensity device) versus frequency R:Z*FREQ as the abscissa (x-axis), here * stands for the DSR module of interest (1—11). Set the frequency and intensity range limits for each axis and start the plot.
2. From the E50 UTILITY hidden window, click on "-Set FRG Scan", a pop-up window will show up. Enter the start scanning frequency at " –Init Freq xxxxx Hz", and end scanning frequency at "-Finl Freq xxxxx Hz" as they are set in

previous step. Also enter step size of frequency scanning at "-Step size xxxxx Hz".

3. Clicking on "*Scan" will start the frequency scanning. Meanwhile, it will also show the status of the scanning operation with "running" in red. The progress of the scanning will be displayed on the FTP plot windows. When the scanning is finished, it will show the status with "stopped" in green.
4. The end user can click on the "*stop" to stop the scan prior to its completion.

Parameter Pages

E50 for ECOOL BPM system is designed for the end users to collect beam position measurements for both the electron and circular beams, and other relevant operations. To the ECOOL BPM system, E50 lets the end user to tell the system to do jobs of their desire. However, the most of the expected results will be shown as the relevant Acnet devices displayed on the parameter pages. The following is a brief description of the Acnet devices we use the most.

For the operational system we maintain several parameter pages, two general control pages, a sample of which is shown in Figure 5, one parameter page for each DSR module, an example is shown in Figure 6, and a parameter page for the electron cooling positional damper, shown in Figure 7.

```
R117 ECBPM CRATE 1 CONTROL          SET      D/A    A/D   Com-U ◆PTools◆
-<FTP>+ *SA◆ X-A/D  X=TIME       Y=T:TULPHT,T:TULAVT,T:TULAHT,T:TULPVT
COMMAND ---- Eng-U  I= 0         I= .575    , .575    , .575    , .575
-< 8>+  Once AUTO   F= 10        F= .595    , .595    , .595    , .595
pelletron  operation  steerers   DIAGNOSTIC  focusing  utilities

 ! FREQUENCY TUNING PARAMETERS

-R:VCFREQ     ECBPM Circ. Freq.      89762        89762      HZ
-R:VEFREQ     Electron Freq.         32050        32050      HZ
-R:VCCALF     Circular cal freq.     89812        89812      HZ
-R:VECALF     Electron cal freq.     32050        32050      HZ

 ! PREAMP GAIN (0) LOW, (1) HIGH
-R:VPAMPG     EC BPM preamp gain     1            1

 ! PULSED MODE OPERATION:
 !  (Y) ON: START PULSED MODE
 !  (N) OFF: STOP PULSED MODE
 !  PARAMETER IS DELAY IN US FROM TRIG TO PULSE
-R:VPULSE     Pulse mode             95           95            N

 ! ERROR LOGER CONFIGURATION
 !  (1) LOG_INFO, (2) LOG_STATE, (3) LOG_DEBUG
-R:VECLRC     EC_LOG Level           0            0          LVL#

 ! REBOOT DSP (VDSPLD) OR CRATE (VSBOOT)
 !  VDSPLD (0) DEV, (1) TEST, (2) PROD
-R:VDSPLD     Dsp Load Level         0            0
-R:VSBOOT     System Reboot          0            0          LVL#

 R:VECMOD     ECBPM Oper. Mode                    1

 ! BPM CALIBRATION
 !  AMPLITUDE IN V (BPCAL), FREQ. IN HZ (BPCALF)
-R:BPCAL      Ecool BPM Calibration   66 -54      66 -54      66 -54
-R:BPCALF     Ecool BPM Cal. freque   66 -54      66 -54

-R:VCBPMS     ECBPMS R39 DATA SETTI  1
-R:VCBPMS[1]  ECBPMS R39 DATA SETTI  0
```

Figure 5: ECBPM parameter page for overall control of the system. A similar page, sub-page 9, exists for the ECBPMA system.

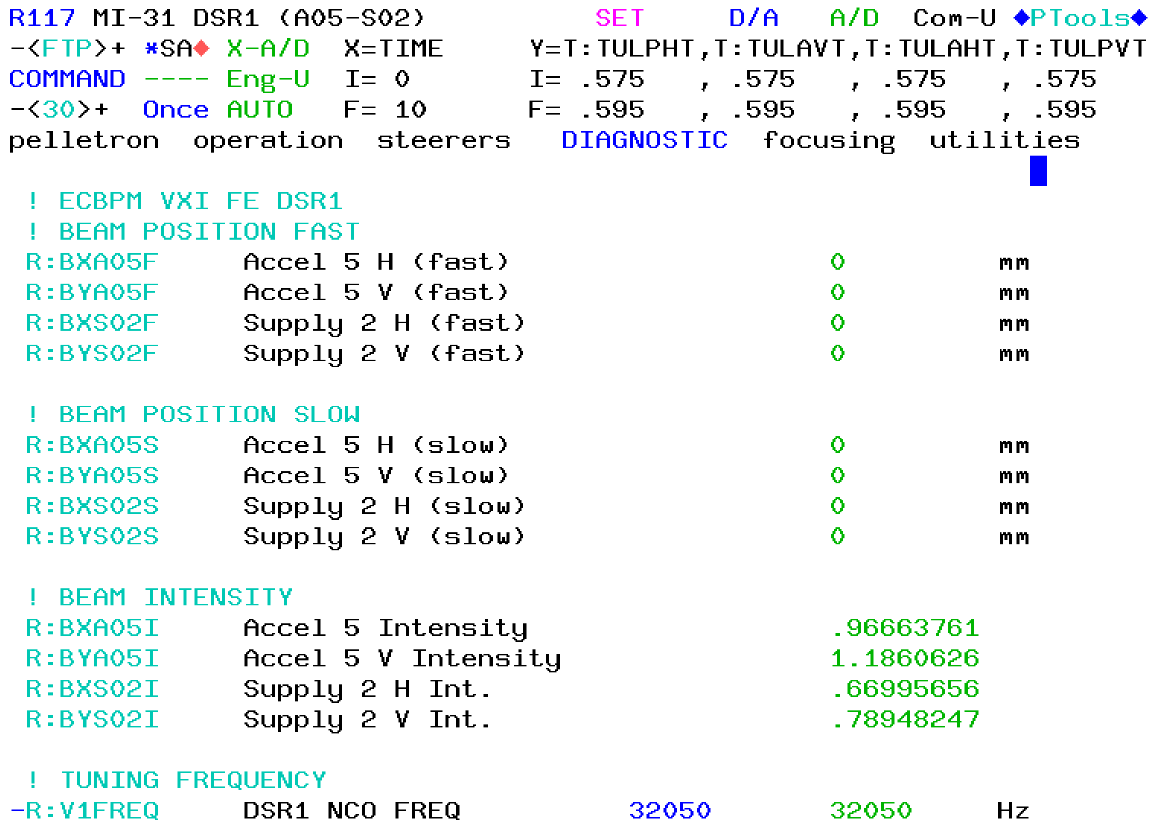

Figure 6: Generic parameter page organization for a given BPM pair in the system at MI-31

```
R117 MI-31 DSR6A (DAMPER)          SET      D/A     A/D   Com-U ♦PTools♦
-<FTP>+ *SA♦ X-A/D  X=TIME       Y=T:TULPHT,T:TULAVT,T:TULAHT,T:TULPVT
COMMAND ---- Eng-U  I= 0         I= .575    , .575    , .575    , .575
-<46>+  Once AUTO   F= 10        F= .595    , .595    , .595    , .595
pelletron  operation  steerers   DIAGNOSTIC  focusing  utilities

 ! INTENSITIES
 R:BXDC0I     Damper 0 H Int.                        39.851509
 R:BYDC0I     Damper 0 V Int.                        46.393166
 R:BXDC1I     Damper 1 H Int.                        43.203907
 R:BYDC1I     Damper 1 V Int.                        49.579941

 ! DATA VALID POSITIONS
 R:EDDVP1     TPS DataValid 1 X                      .4272413  mm
 R:EDDVP2     TPS DataValid 1 Y                     -.24924994 mm
 R:EDDVP3     TPS DataValid 2 X                     -.27202579 mm
 R:EDDVP4     TPS DataValid 2 Y                     -.42466965 mm

 ! TPS/DAMPER LOOP GAINS
-R:ETPSPG     TPS Prop. gain          1             1
-R:ETPSIG     TPS Integral gain       5000          5000
-R:ETPSBT     TPS Beta Value          .015          .015

 ! FILTERED PID OUTPUT
 R:EPID01     TPS PID Output 1                      0
 R:EPID02     TPS PID Output 2                      0
 R:EPID03     TPS PID Output 3                      0
 R:EPID04     TPS PID Output 4                      0

 ! DAC OUTPUTS
 R:EDAC01     TPS DAC Output 1                      0
 R:EDAC02     TPS DAC Output 2                      0
 R:EDAC03     TPS DAC Output 3                      0
 R:EDAC04     TPS DAC Output 4                      0

 ! DAMPER ON/OFF SWITCH
-R:EDPSWH     Damper ON/OFF           0             0

 ! GLOBAL DAMPER GAIN
-R:EDPGN      Global Damper Gain      0             0
```

Figure 7: Parameter page organization for the transverse position damper system at MI-31.

The other Acnet devices, such as I/Q data, probes, are placed on parameter pages as needed. In this way, we can restrict each DSR to one sub-page. Common utility Acnet devices, such as ()CIC2G, ()CIC5G, ()RCFG and ()CBPMR will be placed on another sub-page.

Listing and Description of Acnet Device Names

In general, the differences between the Acnet devices used for the development system and the operational system amount to no more than a prefix change of Z:E for development devices becomes R:V for operational devices. There are some exceptions to this general rule. Principally, the exceptions occur in the case of positions, both fast and slow and intensities. The Acnet devices for the operational system have been named in collaboration with E-COOL BPM system end-users and are shown in Table 1. The prefix notation ( ) is used where appropriate.

- ()xFREQ : tuning frequency

  where x stands for the index of the DSR modules, which are 1 through 9, then, A and B. **Example:** ()3FREQ is the tuning frequency for DSR module 3

- ()xPRBy: probe to read DSP memory location

  where x stands for the index of the DSR modules, which are 1 through 9, then, A and B. There are four probes for each DSR module, y indicates one of those four probes. It can be 1 through 4. **Example:** ()3PRB2 is the probe 2 for DSR module 3

- ()xyzw: I/Q data read after DSR chip latch

  where x stands for the index of the DSR modules, which are 1 through 9, then, A and B. y stands for the BPM channels which is identified with A though D. There are two legs from each channel, so z can be 1 or 2. There are I and Q components for each leg, so w can be I or Q. **Example:** ()1A1I, is the component I for the leg 1 from the channel A in DSR module 1.

- ()DSRID: a settable acnet device, its value is the index of DSR module selected.

- ()CHANX: a settable acnet device, its value can be 1 through 4 for each of four channels. When its value equals 5, it means all of four channels are selected.

- ()xAERy: beam alignment error between electron and circular beams when run in switch mode

  where x stands for the index of the DSR modules, which are 1 through 9, then, A and B. there are to cables connected to two BPM sets for each DSR module. y=1 is for the fist BPM set and y=2, the second one. **Example** ()3AER2 is the beam alignment error for the second BPM set of DSR module 3

- ()PAMPG: Stores the preamp gain setting, 0 is LOW preamp gain and 1 is HIGH preamp gain, has reading and setting properties.

- ()CBPMG: an Acnet array device with only reading properties. It stores the gain settings for the hardware filters on all DSR boards. There are three filters per channel, 4 channel pairs per board and 11 boards per system (max), thus there are a total of 132 array elements.

- ()CBPMC: an Acnet array device with only reading properties. It stores the calibration factors for HIGH preamp gain, both electron beam and circulating beam, for all DSR boards. There are 4 channel pairs per board, 2 beam types and 11 boards per system (max), thus there are a total of 88 array elements.

- ()CBPML: an Acnet array device with only reading properties. It stores the calibration factors for LOW preamp gain, both electron beam and circulating beam, for all DSR boards. There are 4 channel pairs per board, 2 beam types and 11 boards per system (max), thus there are a total of 88 array elements.

- ()CBPMR: a settable acnet device. It reloads different DSP load line to the DSR memory. When its value 0, 1, and represents for the load lines of test, development and production.

- ()START: Stores the scan frequency starting point, has reading and setting properties.

- ()FQEND: Stores the scan frequency end point, has reading and setting properties.

- ()FSIZE: Stores the scan frequency step size, has reading and setting properties.

- ()FSCAN: used to initiate a frequency scan using the parameters stored in the devices ()START, ()FQEND, and ()FSIZE.

- ()PULSE (0x1B): device controlling pulsed mode operation, has reading, setting, basic control and basic status properties. The setting stores the delay in microseconds between a pulse mode trigger and the start of data collection. The control turns pulse mode operation ON or OFF.

- ()ECMODE: Read only device which stores the operational mode of the front-end, 0 = circulating beam (anti-proton or proton), 1 = CW electron beam, 2 = switch mode, alternates between circulating beam and CW electron beam, 3 = pulsed electron beam, 4 = changing between modes.

The following Acnet devices are integrated into the E50 interface, some are settable from a parameter page but the majority are more easily set from E50. The hex value in parentheses following the device name is the channel number for the BPMCLS OID created by the ECOOL system software.

- ()CBPMD (0x00): read only acnet device, used for data caputure from the E50 Data Display window.

- ()CBPMS (0x01): device used to select the operational mode of the system. The mode is usually selected from the E50 PA page and not from a parameter page.

- ()CBPMU (0x02): selects various utility functions from the front end, such as changing filter gains, preamp gains or retrieving calibration factors.

- ()CBPMP (0x03): Stores the detection frequency for the circulating beam, has reading and setting properties.

- ()CBPME (0x04): Stores the detection frequency for the electron beam, has reading and setting properties.

- ()CALFE (0x05): Stores the calibration frequency for the electron beam, has reading and setting properties.

- ()CALFC (0x06): Stores the calibration frequency for the circulating beam, has reading and setting properties.

- ()GCIC2 (0x14): an Acnet array device allowing readback of CIC2 filter gains.

- ()GCIC5 (0x15): an Acnet array device allowing readback of CIC2 filter gains.

- ()GRCF (0x16): an Acnet array device allowing readback of CIC2 filter gains.

Damper specific Acnet devices are shown below. In general, these devices are created only for the second crate (ecbpma) and the development system, as there is a damper system does not exist in the first crate:

- ()DPSWH: Global on/off switch for transverse position damper. A value of 1 indicates the damper is on, 0 is off.

- ()DPGN: Global damper gain for all transverse position damper corrector channels.

- ()TMC1X: On/off switch for transverse position damper channel 1 corrector, currently at position S05X.

- ()TMC1Y: On/off switch for transverse position damper channel 2 corrector, currently at position S05Y.

- ()TMC2X: On/off switch for transverse position damper channel 3 corrector, currently at position B02X.

- ()TMC2Y: On/off switch for transverse position damper channel 4 corrector, currently at position B02Y.

- ()TPSPG: Sets proportional gain for transverse position damper loop.

- ()TPSIG: Sets integral gain for transverse position damper loop.

- ()TPSBT: Sets beta value for transverse position damper loop.

- ()TPSK1: Sets overall gain for transverse position damper channel 1 corrector, currently at position S05X.

- ()TPSK2: Sets overall gain for transverse position damper channel 2 corrector, currently at position S05X.

- ()TPSK3: Sets overall gain for transverse position damper channel 3 corrector, currently at position S05X.

- ()TPSK4: Sets overall gain for transverse position damper channel 4 corrector, currently at position S05X.

- ()TPSM: An array device which stores the transverse matrix elements. These elements can be measured either by the front-end or by a Java controls application. The values displayed are in units of mm/A and when downloaded to the front-end (by an aggregate update on element 0), the values are inverted and loaded to the DSP memory.

Damper Operation

A transverse position damper has been installed in the second BPM crate (ecbpma) to reduce several higher frequency components which induce oscillations in the transverse electron beam position. The major frequency components are 20 Hz, 30 Hz and 60 Hz which can be traced to external mechanical (chain, or shaft) vibrations or electrical noise. The damper is implemented using a extra DSR card to monitor the reflected BPM signals coming from the C00 and C10 locations. These positions are used in a PI feedback loop to eliminate the higher frequency components of the beam motion. The output of the feedback loop is then multiplied by a transfer matrix to enable the proper correction current to be obtained for each of the correctors at S05X, S05Y, B02X, and B02Y. Additional correctors have been installed at these locations for the specific use of the damper system. To turn on the damper, the user should set the database device ()DPSHW to 1, this enables the output. The gain should then be changed for all cahnnesls using the devices ()DPGN. The processing of the loop is shown in Figure 8.

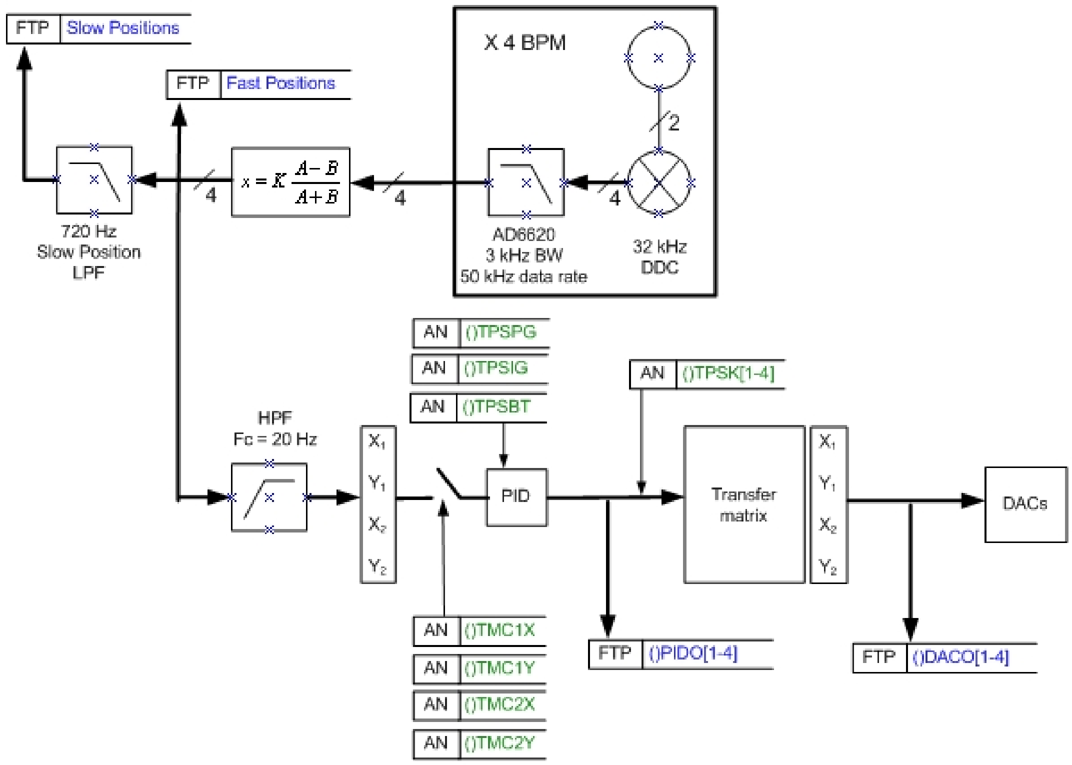

Figure 8: Transverse position damper loop diagram. Settable acnet devices used to tune the loop are shown in blue, while read only devices are shown in green.

Description of hardware

The hardware installed at each location will minimally include the following:

1. VXI crate
2. V152 Slot 0 controller with an installed PMC-UCD card
3. VXI Universal Clock Decoder (UCD) module
4. One or more VXI Digital Signal Receiver (DSR) modules.
5. A 65 MHz clock source for the DSR modules.
6. BPM cabling for input into DSR 15 pin D-type connector
7. A minimum of two TCLK sources, one for the PMC-UCD card and one for the VXI-UCD module.

The number of installed DSR modules will fluctuate over time between 1 and 11 depending upon the changing operational requirements for each station.

BPM

The physical BPM detector contains two 5 cm long split plate assemblies with an inner diameter of 2 inches; One assembly is oriented to collect horizontal position data and the other vertical position data, both are isolated from the BPM housing by ceramic standoffs. A 1/8 inch stainless steel ring, 2 inch I.D., is mounted between the horizontal and vertical detectors and tied to the BPM housing to act as a shield to secondary particle emissions.Two similar shields are mounted at the entrance and exit of the BPM. The present installation at the Wideband laboratory contains two BPM units that have an inner diameter of 2.5 inches at locations (S03 and R03). These will be replaced when the system is installed at MI-31.

Pre-amp

The preamplifier is a cable to a low impedance charge amplifying stage, chosen to minimize the frequency response of the detector and enhance signal to noise. The BPM output signal from each plate is fed to the preamplifier after a short coaxial cable. The amplifier gain can be set by the user at one of two levels, which differ a factor of ten. A differential driver uses the amplified to create a differential signal for each detector plate to eliminate common mode noise between this stage and the next. The differential beam signals are sent through a set of twisted pair cables and detected differentially by a digital signal receiver for further detection and processing.

VXI mainframe and Slot 0 Controller

The ecbpm(d) system depends upon three separate VXI hardware modules, DSR modules to detect and process the BPM signals, a VXI-UCD module to provide beam synchronized timing resources and a "slot 0" controller to coordinate tasks between modules and communicate with higher level protocols, specifically Acnet/MOOC. The DSR module and VXI-UCD module will be described later. All modules are installed into a VXI mainframe conforming to the VXIbus specification containing 13 slots for VXI modules. The VXI mainframe, which conforms to the VXIbus specification, provides a common set of power, trigger and communication lines to all modules through plug-in connectors in the back of the mainframe. This connection block is conventionally referred to as the backplane.

A Kinetic Systems V152 slot 0 controller occupies the first slot (slot 0) in the VXI mainframe. This module combines an onboard PPC based CPU and memory with VXI interface electronics. The slot 0 controller performs many functions, including initialization, code storage and execution, and communication internally within the mainframe and externally across an Ethernet network. The slot 0 controller acts as the hardware interface between the end user application, E50 PA, and the low level data processing hardware, DSRs and BPMs.

DSR

The Digital Signal Receiver (DSR) module is a size C VXI module composed of eight digital receiver channels each containing a 14 bit/65 MS/s ADC (AD6644) and a digital down conversion chip (DDC) with three separate filters (AD6620). The down conversion frequency and filter gains can be changed dynamically during system operation. The impulse response function of the last hardware filter in the signal chain may be redesigned and loaded at boot time or reloaded upon a configuration change. Both the ADC and DDC chips were obtained from Analog Devices™. Each DSR board also contains one SHARC* DSP chip, also from Analog Devices, which provides on-board processing capabilities at a clock rate of 40 MHz. The incoming signal is digitized at 65 MHz, down converted and filtered, and then passed to the DSP for further processing. The signal input to the DSP is passed through a software filter and then used to calculate the beam position in mm using the sum over difference formula

$$x = K_{scale} * \frac{A - k_{cal} B}{A + k_{cal} B} + K_{offset}$$

where *A* and *B* are the signal intensities observed by each plate of a BPM pair, K_{scale} and K_{offset} are scale and offset factors determined by the physical construction of the BPM, and k_{cal} is a calibration factor which accounts for small differences in signal path gain between *A* and *B* plates.

In the BPM application, the DSR channels are grouped into pairs, each pair corresponding to an orientation, vertical or horizontal, for a particular BPM location. A single DSR card can gather data from 2 BPM locations, or a total of 4 plate orientations. A full crate containing 11 DSR modules can thus support no more than 22 BPM locations.

VXI-UCD

A VXI-UCD module is a multi-purpose timing and decoding resource used by front-end applications to detect accelerator events or decode accelerator data. The VXI-UCD accepts Tevatron ClocK (TCLK), Machine DATa (MDAT) and Beam Sync clock (BSclk) inputs; the ECOOL BPM front-end does not use MDAT or BSclk signals. The front-end programmer can attach application specific signals to one or more detected events. The UCD module may generate a variety of signals, including IRQs, TTL triggers and ECL triggers. The output pattern is also adjustable, the width of a TTL triggers and its delay from an event can be specified using the Pulse Width and Delay feature of the UCD.

In the ECOOLBPM system the VXI-UCD is used to detect event 0xEB. This event signals the generation of an electron pulse inside the Pelletron. After detecting 0xEB, the UCD module is configured to generate a TTL pulse on TRIG line 2 after a fixed delay. The TTL trigger is written or sourced on the backplane as well as the front-panel LEMO connector. The backplane trigger is available to all DSR modules enabling them to synchronize their data collection in pulsed mode to the electron beam generated. There is a finite data processing delay between the receipt of the trigger pulse and the appearance of signal at the DSR processing block. This delay is empirically determined and can be set using the Acnet device ()PULSE, the units are microseconds.

An additional UCD card, a PMC-UCD, is installed in the V152 Slot 0 controller to provide timing resources exclusively for MOOC communication; the ECOOLBPM software does not explicitly use this module.

---

* Super Harvard Architecture Chip™ Analog Devices.

Software

Overview

The ecoolbpm front-end project along with the supporting DSR shared library code and ecbpmdsrdsp DSP project code form the core system software of the ECOOL BPM system. Other rfies shared libraries, vucdrfies, an++, anmemarr, error, as well as Controls department MOOC software provide additional functionality to the system. We will attempt to give a brief overview of the code and its operation, with subsequent sections addressing the ecoolbpm front-end project software (code) and the ecbpmdsrdsp DSP project code.

The ECOOL BPM system software, ecoolbpm and DSP code, provides all necessary functionality to support end-user requests for BPM measurements within the ECOOL BPM operational system. Its duties include:

- Identifying the hardware available in the VXI mainframe, creating a software interface to the hardware and configuring the hardware in a known state.
- Managing communication to the hardware modules to perform end-user requested changes in configuration.
- Configuring the operational mode of the software in a initial know state, and changing the software and hardware configuration to support separate operational modes.
- Calculating BPM data, positions and intensities, from hardware acquired raw data signals.
- Initializing communication with Acnet, exposing system variables (positions, intensities) of interest to end-users and linking user specific work requests to software tasks or functions.
- Managing communication between end-users and the system to respond to work requests in a timely and consistent fashion.
- Reporting errors either to console PA or terminal windows for system configuration failures.

We make a distinction between software that is loaded and running on the V152 Slot 0 controller and software that is loaded onto the Analog Devices ADSP-21062™ chip on the DSR module. The first set of software, collectively referred to as slot 0 code, includes the ECOOLBPM system software and any additional supporting libraries compiled under VxWorks. The second set of software, DSP code, is loaded onto the DSP hardware chip using slot 0 software function calls when the system is initialized. These two different sets of software have specific roles within the ECOOLBPM system.

The DSP code handles raw data processing, exposes system specific information and manages the low level communication with the hardware installed in the DSR module. The data processing element consists of extracting the data from hardware registers, applying additional filtering, and calculating system specific information, positions and intensities. The low-level communication element consists of carrying out requests from slot 0 software to perform specific hardware configuration or DSP software mode/state changes. These requests follow a LLRF-ESG standardized client-server protocol where the slot 0 controller is the *client* that requests the DSP software, the *server*, perform work. The DSP calculates intensities and positions and places these values in a fixed memory location; these are then mapped to Acnet by the slot 0 software. For more information on the DSP software you are referred to the DSP section of this document.

The slot 0 code acts as glue layer or middleware between the end-user requests and the low level data processing. It is responsible for performing all necessary work to configure the system in response to user requests for BPM system information. This includes initialization, re-configuration, and establishing and managing communication with Acnet database devices, requesting specific work from the DSR as a DSP software *client.* The slot 0 software will be dealt with in more detail in the front-end slot 0 software section.

Common operational modes

The ECOOLBPM system supports four high-level operational modes:

1. Circulating beam measurements[†]
2. Electron beam measurements
3. Pulsed electron beam measurements
4. Switch or alternating mode

The primary difference between these modes is the frequency at which the DSR module will detect the incoming signal. For circulating beam measurements the detection frequency is the revolution frequency of the recycler. For electron beam measurements, the detection frequency is the modulation frequency of the electron beam. The switch or alternating mode, alternates between circulating beam measurements and electron beam measurements at a specified frequency. These three modes measure DC or CW beam. Pulsed electron beam measurements detect a short (~ 2 μs) electron pulse, the FFT of such a pulse gives a broad frequency domain spectrum and thus detection frequency is not as critical in this mode. The time between the generation of the pulse and the triggering of data collection within the software is critical. This value can be set using the acnet device ()PULSE.

The operational mode can be changed from E50 using the data display window. Selecting the beam type for CW measurements and interrupting or clicking on snapshot reconfigures the system for the new mode. Interrupting or clicking on PULSE reconfigures for pulsed mode operation. Reconfiguration of operational mode is performed only once, in other words the software retains knowledge of its operational mode and will not reconfigure for the same operational mode.

Operational mode reconfiguration and other end-user requests of the ECOOLBPM system cause the slot 0 controller to request one or more DSRs to perform work on its behalf. The slot 0 controller software acts as a client and requests services from the DSR servers by issuing vector interrupt commands to perform specific functions, change

[†] Circulating is defined here as beam circulating within the recycler.

detection frequency, change gains, or reload filter files. As an example take the case of changing the detection frequency to support CW electron beam measurements. The flow of control proceeds as follows:

1. The slot 0 software invokes functions from the DSR shared library to request that each DSR change its down conversion frequency to the requested value.
2. The DSR library function validates its arguments and then transfers those arguments from slot 0 memory to DSP registers.
3. The DSR library function then writes a DSP memory address (the "vector interrupt") to the DSP vector interrupt register (VIRPT).
4. If the DSP VIRPT register is non-zero the DSP software immediately jumps to the address written in that register.
5. At the new address, one of a series of VIRPT address organized as a jump table, a second jump instruction forwards the call to the appropriate DSP routine to perform the work.
6. The DSP processing routine extracts the arguments from the registers and performs whatever processing it needs to do to change the detection frequency for that particular DSR.
7. Once this processing is finished the DSP routine writes 0 to the VIRPT register to indicate it has completed its work. Errors during the processing are written to a separate DSP register.
8. The slot 0 software continually polls the VIRPT register to determine when the work it requested has finished and it checks the error value to ensure the work was done properly. The slot 0 software can then proceed with other requests.

Front-end slot 0 code

Initialization

The ecoolbpm system is a diskless embedded system, which attaches to a remote network host via Ethernet to download its OS (VxWorks) and operational software. The boot process is controlled by a set of reconfigurable boot parameters, which are shown below, held in internal memory on the V152 slot 0 controller. These parameters specify network parameters, IP addresses of the ecoolbpm node, IP address of the boot host (fecode-bd), the Acnet trunk and node values for the system, and the location of the OS kernel image, operational software and startup script on the boot or download host.

Boot parameters

```
ECBPM

boot device          : dc0
processor number     : 0
host name            : fecode-bd
```

```
file name            : vxworks_boot/fe/ecoolbpm/v152/vxWorks
inet on ethernet (e) : 131.225.134.199
inet on backplane (b):
host inet (h)        : 131.225.121.145
gateway inet (g)     :
user (u)             : vxworks_boot
ftp password (pw) (blank = use rsh):
flags (f)            : 0x0
target name (tn)     : ecbpm_0xbb6
startup script (s)   : /vxworks_boot/fe/ecoolbpm/v152ecoolbpmstartup
```

ECBPMD

```
boot device          : dc0
processor number     : 0
host name            : fecode-bd
file name            : vxworks_boot/fe/ecoolbpmd/v152/vxWorks
inet on ethernet (e) : 131.225.23.48:ffffffc0
inet on backplane (b):
host inet (h)        : 131.225.121.145
gateway inet (g)     : 131.225.23.62
user (u)             : vxworks_boot
ftp password (pw) (blank = use rsh):
flags (f)            : 0x0
target name (tn)     : ecbpmd_0xb59
startup script (s)   :vxworks_boot/fe/ecoolbpmd/v152ecoolbpmdstartup
```

Startup Script

The startup script executes after the OS kernel image has been downloaded onto the system and performs the following tasks to bring the system to a default or base operational state.

1. Mount relevant download host directories with simple aliases.
2. Load MOOC and Acnet code and configure the PMC-UCD card for operation.
3. Load ecoolbpm supporting libraries, liberror, liblog, etc.
4. Load ecoolbpm system software, libdsr, libecoolbpm.
5. Run the resource manager to search through the VXI mainframe for all installed VXI modules.
6. Create an address book for system object storage and retrieval.
7. Create and error logging object.
8. Initiate MOOC communication between ecoolbpm and Acnet.
9. Set the values of two global variables
    a. systemType – specifies operational (1) or development (0) system
    b. DspLoadLine – specifies what level to load DSP code from
10. Run EcMain, the main entry point for ecoolbpm system software

EcMain

`EcMain` is the main entry point for the ecoolbpm system software, the first action of the routine is to call `EcNew`, the system wide constructor for all ecoolbpm specific software objects. `EcNew` performs the following tasks:

1. Creates system error loggers.
2. Creates AnIntFlt devices for Acnet database communication.
3. Adds callback functions for these Acnet devices.
4. Allocates space for application data structures.
5. Identifies installed DSR modules, creates software objects to represent each DSR and downloads DSP code to each module.
6. Creates an AnMemArray instance to access the FTP variables for each installed DSR module.
7. Creates a software object to represent a VXI-UCD module.
8. Creates and event manager object.

Once all software objects are constructed, control returns to `EcMain` and `EcConfig` is called to establish a default system configuration. `EcConfig` reads previous values of the calibration factors and loads them into the DSP memory, configures the VXI-UCD module to detect TCLK event 0xEB and assert trigger line 3, and reads the model code from each DSR and writes this to an Acnet device. Errors in EcConfig will be logged to the console and cause an abnormal termination leaving the system in an unconfigured state. After EcConfig returns, EcMain assigns values to several global variables, creates a MOOC class (BPMCLS) for E50 communication and control and initiates an auto-download of registered Acnet database devices. Once this is completed the system exists in a useable state and is put into an idle mode where it waits for events, requests from end-users.

Organization

A brief breakdown of the file organization and function location is given in the following two tables:

Table 5: Program code files included in the ecoolbpm front-end project and a brief description of their contents.

File	Purpose
ecancllbk.c	Acnet callback routines to support application specific database device settings and basic control through the anIntFlt and anMemArray classes
ec.c	Routines to construct and support basic MOOC class communication, callback assignment and invocation between console PA and front-end
eccalib.c	Routines to support calibration of the signal conditioning chain
ecconfig.c	Routines to configure the ecoolbpm front-end in a known default state
ecdump.c	Routine to display information to a console terminal
ecgain.c	Routines to set and retrieve hardware filter gain values
echelp.c	Routine to display ecoolbpm function interface
ecmain.c	EcMain, and routines to support high-level operational modes

ecnew.c	Ecoolbpm software object constructors and helper functions
ecpreamp.c	Routines to set preamp gains and scale
ecstatus.c	Routines to display messages to E50 Message window
ecutil.c	Routines to support reloading DSP code and other utility functions

Table 6: File and function reference for ecoolbpm software.

File	Function	Purpose
ecmain.c	`EcMain`	Main entry point for ecoolbpm software
	`EcAproton`	Configures the system to measure the position of circulating beam
	`EcElectron`	Configures the system to measure the position of the electron beam
	`EcSwitch`	Configures the system to alternate between measuring the position of the electron beam and the circulating beam
	`EcPulse`	Wrapper function to switch to pulse mode
	`EcPulseOn`	Turns pulse mode on for all DSRs
	`EcPulseOff`	Turns pulse mode off for all DSRs
	`EcDsrName`	Assigns global names for DSRs and ANDSR instances
	`EcAlignErrorData`	Calculates alignment error between beams
ecnew.c	`EcNew`	System public constructor
	`EcCreate`	System private constructor
	`EcGetDspInfo`	Gets information on DSP FTP variables and creates acnet AnMemArray access
	`EcGetAlignError`	Creates AnMemArray access to alignment error parameters
	`EcCllbkAdd`	Adds callback functions for anFltId
	`EcDspMap`	Maps DSP FTP variable data to E50 data structure
	`EcTestLoadError`	Diagnostic function to test the loading of DSP code
ecancllbk.c	`EcAnDSRIDCllbk`	Callback to set DSR index
	`EcAnCHANXCllbk`	Callback to set Chan index
	`EcAnFREQCllbk`	Callback to set NCO frequency of a single DSR
	`EcAnSCANCllbk`	Callback to initiate frequency scan
	`EcAnSPHASCllbk`	Callback to set the NCO phase of a single DSR
	`EcAnCIC2GCllbk`	Callback to set the CIC2 filter gain of

		a single DSR
	EcAnCIC5GCllbk	Callback to set the CIC5 filter gain of a single DSR
	EcAnRCFGCllbk	Callback to set the RCF filter gain of a single DSR
	EcAnFREQ1Cllbk	Callback to set the start frequency for a frequency scan
	EcAnFREQ2Cllbk	Callback to set the stop frequency for a frequency scan
	EcAnSTEPSCllbk	Callback to set the number of steps for a frequency scan
	EcAnSFREQCllbk	Callback to set the NCO frequency for a frequency scan ??
	EcAnCllbk	Generic callback function for an int/float setting callback functions, copies setting to reading
eccalib.c	EcCalibOne	Calibrate a single DSR
	EcGetCalib	Get calibration constants for all DSRs from DSP memory
	EcGetCalibOne	Get calibration constants for one DSRs from DSP memory
	EcSaveCalib	Saves calibration constants for all DSRs to disk file
	EcReadCalib	Read calibration constants for all DSRs from disk file
	EcLoadCalib	Download calibration constants for all DSRs to DSP memory
	EcShowCalib	Display calibration constants for a given DSR
	EcSetCalibM	Set all calibration constants to a given value
	EcSetCalibMx	Set calibration constants to a given pattern
	EcOnLineCalib	Calibrate all DSR modules
ecconfig.c	EcConfig	Configure hardware components for operation and download data files
ecdump.c	EcDump	Display ECOOLBPM object information to console
ecgain.c	EcSetGain	Set new filter gains for all DSRs, write to Acnet and save to file
	EcSetGainX	Set new filter gains for all DSRs, write to Acnet, DO NOT save to file
	EcSetGainM	Set filter gains to minimum value, DOES NOT not transfer to hardware
	EcGetGain	Get filter gains for all DSR modules

		from DSP memory
	EcGetGainM	Get filter gains for one DSR module from DSP memory
	EcSaveGain	Saves filter gains to disk file
	EcReadGain	Reads filter gains from disk file
	EcLoadGain	Transfers filter gains from slot 0 memory to DSP memory
	EcShowGain	Display filter gains for a given DSR to console
	EcShowGainAll	Display filter gains for all DSRs to console
	EcUpdateGain	Update gains for E50??
echelp.c	EcHelp	Prints inteface of ecoolbpm to console
ecpreamp.c	EcSetPreampScale	
	EcSetPreamScaleM	
	EcSavePreampScale	
	EcSavePreampScaleM	
	EcReadPreampScale	
	EcLoadPreampScale	
	EcSetPreampGain	
	EcSetPreampScaleOne	
ecstatus.c	EcSetState	Passes an indexed message to E50 message display window
ecutil.c	EcLoadOffsetScale	Transfers BPM specific scale and offset values to DSP memory
	EcSetPulseTrigDelay	Sets delay for pulse mode for all DSR modules
	EcResetDsr	Reboots the DSR modules and loads new DSP file from a given level (PROD, TEST, DEVEL)
	EcFreqScan	Performs a frequency scan
	EcDsrReboot	Reboots the DSR modules and loads new DSP file passed as a parameter
ec.c	EcReadD	MOOC reading callback function
	EcSetD	MOOC setting callback function
	EcReadS	MOOC read of setting callback function
	EcBasicCTL	MOOC basic control callback function
	EcBasicSTS	MOOC basic status callback function
	EciSetCalib	Writes calibration constants to Acnet devices
	EciSetGain	Writes DSR filter gains to Acnet devices
	EcOpMode	MOOC callback to set operational

		mode of system
	`EcUtility`	MOOC callback to set perform several utility tasks, setting gains, changing frequency, starting frequency scans
	`Ecinit_class`	Initializes MOOC class and installs callback functions.

DSP code

Initialization

DSP code is stored on the fecode-bd fileserver and downloaded onto each DSR board using the DSR shared library function DsrLdr(dsrId, filename), where dsrId is the object instance pointer to a DSR and filename is a relative path to the DSP code file. Following downloading of the DSP code, a default filter configuration is written to the AD6620 DDC and the operational state of the software is configured by default to measure the DC electron beam at a frequency of 32 kHz at a high preamplifier gain setting. Some of this configuration is done by the slot 0 code through vector interrupt requests to the DSR boards.

Basic processing modes

There are two basic processing modes that the ECBPM DSP code supports which we call normal mode and pulsed mode. Flow charts describing these modes and the routines used in their execution are shown in Figure 9 through Figure 12. Normal mode is the default configuration after reboot. Briefly, the DSP receives DataValid interrupts at a rate equal to the sample rate divided by the product of the decimation rates for the three filters in the AD6620 chip. Once a DataValid interrupt is received, the software retrieves the I/Q demodulated data and applies a 100 Hz filter to it in assembly code. After a fixed number of DataValid interrupts (~ 720 Hz) rate a C routine, `DataProcessess`, is initiated which retrieves this filtered data, calculates intensity and position for each of the four channel pairs. The positions are further filtered to a 5 Hz bandwidth and positions and intensities are stored in memory exposed to Acnet through database devices. Within pulsed mode the processing is similar, however the assembly level filtering of the data is bypassed.

Intensity and Position Calculations

Each BPM channel, horizontal or vertical, receives signals from 2 pick-up plates. These signals are directed to the inputs of 2 separate channels on a LLRF DSR card. This pair of channels are conventionally referred to as A and B. Within the signal chain for a given DSR channel (A and/or B) the signal is sent through an I/Q demodulator which down-converts the signal to baseband. This creates a vector representation of the signal (I_A , Q_A) and (I_B , Q_B). Each component can be amplified or attenuated identically at three separate hardware filtering stages. The gains applied here are used to ensure the processed signal does not saturate and, as a result, gives reasonable and meaningful values for the calculated position. The possible gain settings are specified in steps of 6 dB, a factor of 2 change in the signal, between specified limits. The minimum values for these gains, the default values used for continuous mode, are given in the following table:

Table 7: Minimum gain settings in dB for AD6620 hardware filters.

Filter	Minimum Gain / dB
CIC2	-36
CIC5	-120
RCF	-18

The gains for each channel of the pair are set identically. In order to reflect the true intensity of the signal as seen by the BPM, the acnet parameters R:BXA01I, R:BXS0*I, etc. are normalized using the filter gain values according to the following algorithm.

First the total ``un-normalized" intensity for the BPM orientation is calculated. The signals from each plate, after passing through the hardware filters, are further filtered using a unity gain software filter. These intensities are then combined using the standard formula:

$$I_{A,tot} = \sqrt{I_A^2 + Q_A^2}$$

and identically for the B signal chain. These single channel intensities are then averaged to obtain a value for the "total" intensity detected by that BPM orientation (i.e. horizontal or vertical).

$$I_{tot} = \frac{I_{A,tot} + I_{B,tot}}{2}$$

the value of I_{tot} is reported in the Acnet devices R:BXA04I, R:BXS02I, and equivalent devices. The beam position in mm is calculating using a cubic expansion in the sum over difference of plate intensities

$$r = \frac{A - k_{cal}B}{A + k_{cal}B}$$

$$x = K_0 + K_1 r + K_2 r^2 + K_3 r^3$$

where *A* and *B* are the signal intensities observed by each plate of a BPM pair, K_0, K_1, K_2, and K_3 are polynomical coefficients determined by the physical construction of the BPM, and k_{cal} is a calibration factor which accounts for small differences in signal path gain between A and B plates.

VIRPT processing

The slot 0 controller must be able to communicate with the DSR modules in order to request services. These services are in addition to the processing already being performed by the DSP, i.e. beam position calculations. The slot 0 controller requests services from each DSR using the DSP vector interrupt (VIRPT) register. When the front end is started or rebooted "object's" representing the DSR modules in the crate are instantiated within the slot 0/front-end software. These object references encapsulate module specific information, including a memory-mapped image of the IOP registers within the DSP for that module. To request a specific function from the DSR, the front-end code employs the following "DSP function call" mechanism:

1. Write any parameters needed by the DSP function into one or more of the DSP IOP message registers, `MSGR0`, `MSGR1`, …, `MSGR7`.
2. Write a defined value to the `VIRPT` register to invoke the function. The value is a jump table address tied to the DSP specific function implemented in assembly code.

Once the front-end writes a non-zero value to the `VIRPT` register, a vector interrupt is signaled and the DSP, after first pushing the status stack, jumps to the address written to the `VIRPT` and begins executing the code it finds there until it detects a return from interrupt command (`rti`) . The message registers may also used to return values to the front-end. Conventionally, `MSGR0` stores the return result from the vector interrupt ISR.

In this situation, the interrupt service routine invoked by the vector interrupt would be required to start at some fixed address in the DSP memory space, specifically the address the front end writes to the `VIRPT` register. If the VIRPT code were moved, the address the front-end needs to write to the `VIRPT` register would need to change accordingly.

Jump Table

In order to separate the explicit memory placement of DSP VIRPT routines from invocation of that routine we use a fixed set of vector interrupt values within the front-end code but independently place the particular VIRPT within DSP memory. To accomplish this we introduce a jump table to link the front-end code to the vector interrupt routines. A small, fixed block of DSP memory is allocated to store the jump table. The front-end code defines the addresses of the jump table as a set of constants tied to VIRPT routines. Executing a VIRPT routine proceeds as described previously, write the value defined for the desired routine to the `VIRPT` register. Now, instead of jumping to the first line of the VIRPT routine the DSP jumps to an address within the jump table. At that address a single instruction exists, a jump instruction to the assembly label for the VIRPT routine requested. For example, the front-end wants to change the down conversion frequency through a VIRPT with the assembly label SetNCOFreq. The programmer would implement and store the assembly code for this routine anywhere he/she saw fit. In the jump table, an entry would be inserted as follows

```
jump SetNCOFreq;
```

The exact location of this instruction in the jump file is then determined; in this example we'll assume it occurs at an address of `0x24020`, the following line would then be added to the front-end shared library private header filR:

```
#define NCO_SET_FREQ    0x24020
```

In order to execute the VIRPT routine, the value `NCO_SET_FREQ` is written to the `VIRPT` register. Conventionally, the jump table is stored between address `0x24000` and `0x2403f` in DSP memory.

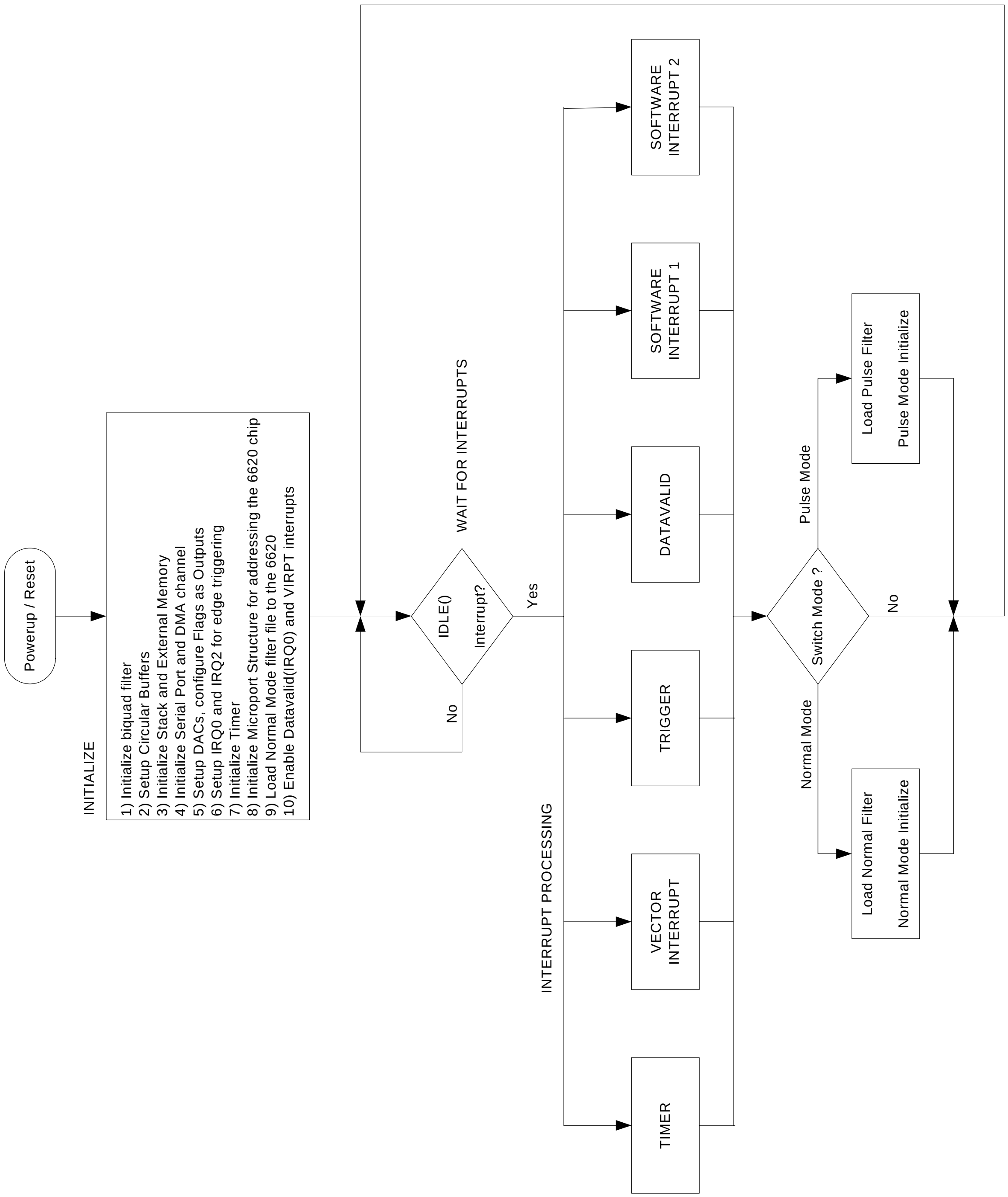

Figure 9: ECBPM DSR DSP Main loop processing flowchart.

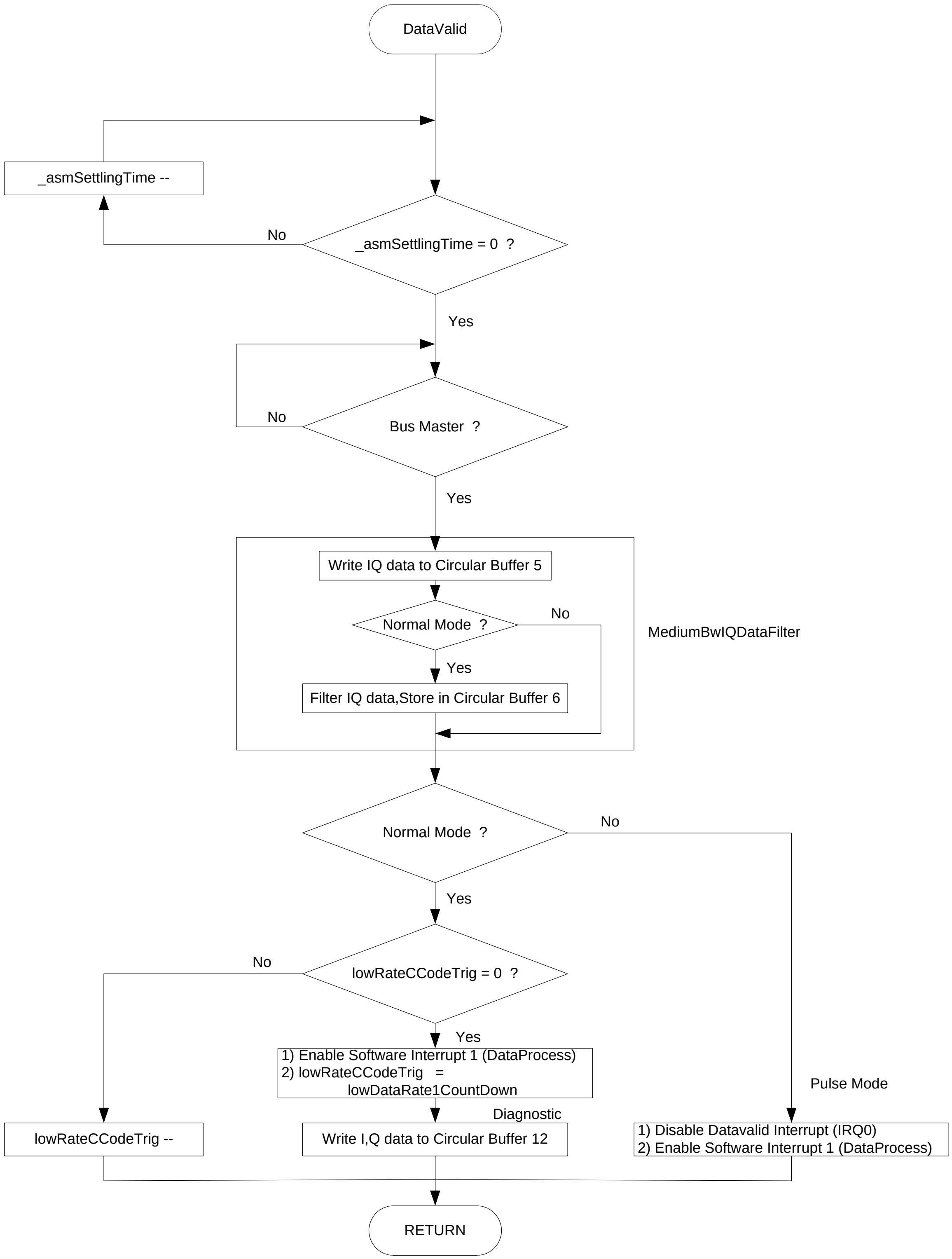

Figure 10: ECBPM DSR DSP DataValid interrupt processing flowchart.

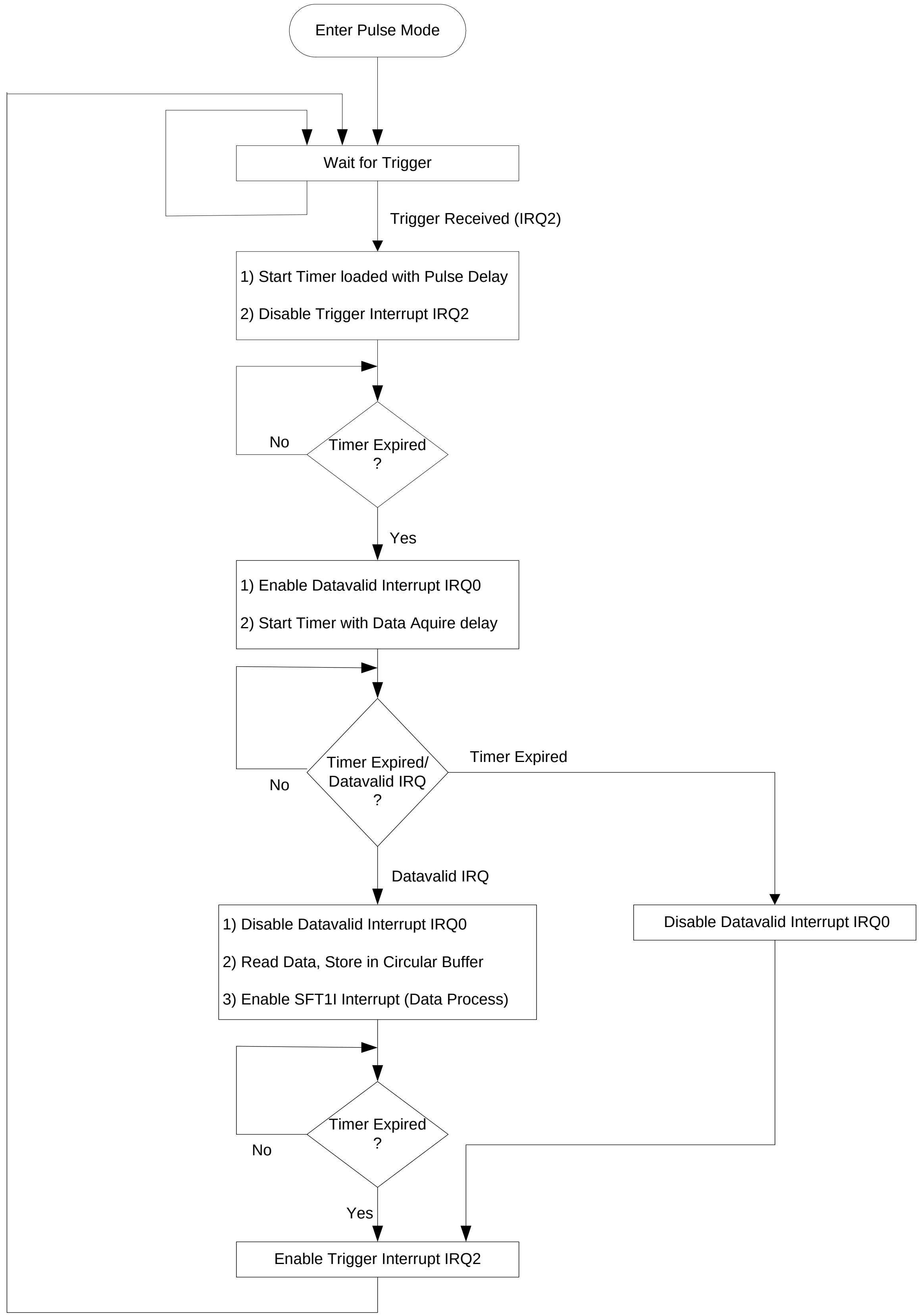

Figure 11: ECBPM DSR DSP Pulse Mode processing flowchart.

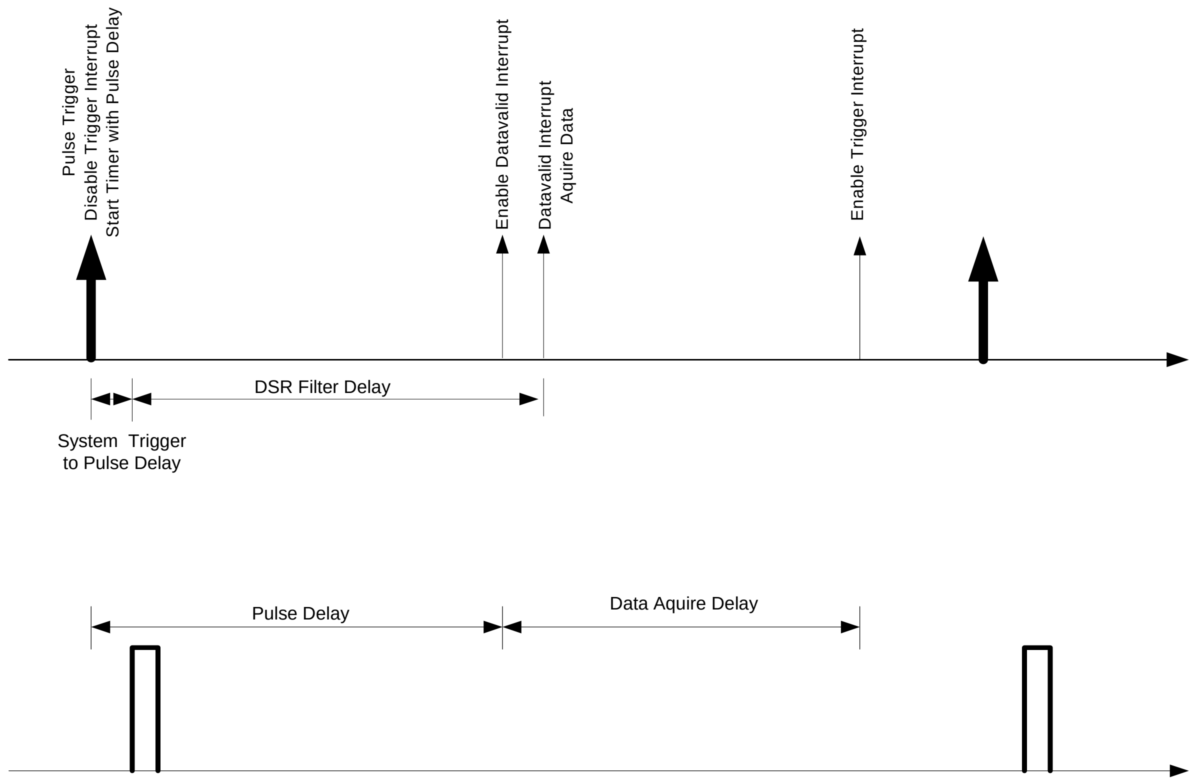

Figure 12: ECBPM DSR DSP Pulse Mode timing diagram.

Troubleshooting

DSR Test Suite (TESTDSR)

In order to troubleshoot the VXI DSR module, a test front-end project was developed to exercise the hardware of the module, this suite resides in the `testdsr` front-end project and the supporting `dsrtestdsp` DSP project.There are five distinct test modes available in the `testdsr` project. They arR:

1. **ADC Tests**: Used to determine if the ADC for each DSR channel is working properly. Data from the I or Q data latches may be selected in this test mode. Data is plotted in a LabView client.
2. **Memory Tests**: Used to determine if the links to memory between the front end V152 and DSP chip are free of error. Three different tests are available.
3. **Frequency Sweep**: Used to sweep the frequency of a given DSR channel over four different user selectable ranges. Data is plotted in a client.
4. **Trim Potentiometers**: Allows dynamic adjustment of the potentiometers to zero any DC offset terms. Data is plotted in a LabView client.
5. **Power Sweep**: Sweeps the input power of a given DSR channel pair over a given range. Used to determine how the calculated position changes as power changes. Data is plotted in a LabView client.

In order to begin testing a particular DSR module the following procedure is recommended.

TestDsr Hardware

Assemble the following hardwarR:

- Wavetech Generator - or other signal generator, for use as signal input.
- Hewlett-Packard E4134B - used for clock input and power sweep generator.
- NIM 53 MHz oscillator} - used for clock input.
- DSR Module - module being tested.
- SWH Module - for triggered output to power sweep generator.
- V152 Module - slot 0 controller for the tests.
- Computer with LabView software - used for control and data viewing for ADC tests, frequency sweep, trim adjustment and power sweep.

Connect the front panel TRIG 0 LEMO connector to the EXT TRIG input on the HP E4134B. Cabling and connectors for the DSR will be needed and these will differ depending on the front panel version (ECOOL or LLRF). Boot the V152 in a VXI crate using the boot parameters shown belowalong with the given DSR module.

V152 Boot Parameters

The boot parameters for the v152 module should be set as shown in Table 4. The `testdsr` program suite must be run in a location which has access to Fermilab subnet 23, i.e. LLRF VXI lab or nearby.

Table 8: Boot parameters for V152 slot 0 controller necessary to run `testdsr` code.

Parameter	Value
boot device	dc
unit number	0
processor number	0
host name	fecode-bd
file name	vxworks_boot/fe/testdsr/v152/vxWorks
inet on Ethernet	131.225.23.200
host inet	131.225.121.145
user	vxworks_boot
flags	0x0
startup script	vxworks_boot/fe/testdsr/v152testdsrstartup

The startup script initially places the DSR board in the ADC test mode. Further details on specific test modes are given below. The channel numbers referenced in the ADC, frequency scan, and trim tests refer to "front panel" single channel numbers, running from 1 to 8. The power sweep test requires a pair of channels, the convention adopted labels the channel pairs with letters shown in Table 5.

Table 9: Channel naming conventions for DSR module channel pairs and mapping to single DSR channels.

Channel Label	Channel Pair
A	1-2
B	3-4
C	5-6
D	7-8

LabView Control

A LabView interface (`TestDsrCTRL`) or virtual instrument (VI) has been constructed to simplify the implementation of tests within the testdsr front end project. The interface of the `TestDsrCTRL` VI was made to resemble the operation of a sequence table (I6, R6, or T6). The parameters for each test are set in the `TestDsrCTRL` VI and then "Send to Hardware" transfers these to the slot 0 controller. If the test mode has been changed, the controller will reset the DSP and load new code specific to the test in addition to updating other parameters.

Once the DSP code has been loaded the test is initiated by the "START TEST" button.

ADC Testing Procedure

Connect the output of the signal generator to the channel of interest on the DSR. Connect the output of the HP E4134B to the CLK IN input of the DSR, set its output to 65.0 MHz, 0.0 dBm, RF ON, MOD OFF. Set the frequency and amplitude of the E4134B to 30.0 MHz and 0.0 Vpp, respectively.

Choose the ADC test mode in the `TestDsrCTRL` VI and choose the channel of interest from the drop down list. Click "Send to Hardware". Load the `TestDsrADC` VI. Initiate the test by clicking "START TEST". Begin the data collection within LabView. You should see a gaussian distribution, if you see a single spike at bin 8192, there has either been a problem loading the filter file (are LEDs 1and 3 on?) or there is a break in the data path for that channel.

Adjust the trim potentiometer of the DSR to obtain an average bin position of 8192 as displayed by the `TestDsrADC` VI. During this adjustment, it is beneficial to periodically restart the histogram by pushing the Restart Histogram in the LabView application.

Next adjust the amplitude of the signal generator to be approximately 2.2 Vpp; this ensures all bins are receiving counts without saturating the ADC. Saturation is visible if the count jumps to high values for bins 0 or 16383. A normal histogram should show an approximate U shaped distribution with high counts near the edges and roughly equal counts in the middle. The red plot in the histogram is the expected value and the width of this distribution should be adjusted to match the width of the actual incoming data; changing the value of Vpp in the LabView application accomplishes this. The differential non-linearity should be random and near 0.

Stop the data collection within the `TestDsrADC` LabView VI by pressing the QUIT button. You may stop the data collection in the DSR by clicking the "Kill Test" button but this is not explicitly necessary. Repeat the measurement for each channel.

Memory Testing Procedure

Choose the Memory test mode in the `TestDsrCTRL` VI and choose the type of test from the Frequency Mode drop down list. The labels explain the options:

- Random values written to DSP memory.
- Address and data bus test of DSP.
- Full device test for a particular region of DSP memory.

Click "Send to Hardware". Initiate the test by clicking "START TEST". Errors are reported to the console. This mode of operation can also be used to clear the DSP memory after a bad previous load of code as initiating this test mode resets the DSP itself clearing the memory.

Frequency Scanning Procedure

Set the signal generator signal generator to 0.0 dBm input at a frequency of 89.0 kHz, place this input into channel 1. Ensure that the clock frequency to the DSR is 65 MHz from the HP signal generator, with amplitude 0.0 dBm.

Choose the Frequency test mode in the `TestDsrCTRL` VI, choose the channel of interest and the particular frequency scan to perform from the respective drop down lists. The possible frequency scans arR:

Table 10: Frequency scans available in the `testdsr` suite.

Frequency Scan	Test Description
0	0 to 100 kHz, in steps of 100 Hz
1	0 to 1 MHz, in steps of 1 kHz
2	52 to 54 MHz, in steps of 1 kHz
3	0 to 32 MHz, in steps of 5 kHz

Click "Send to Hardware". Load the LabView client VI `TestDsrFreq` and start the VI running, right arrow button on toolbar. Initiate the test by clicking "START TEST". During the scan the frequency input into the channel can be changed and a peak should appear at approximately that frequency in the LabView plot. The frequency scale in LabView is only approximate, it is not directly tied to the frequency input. Small errors between ordinate value and frequency should not be taken as indicative of an error. If the error is large or a peak does not occur at all near an input frequency this does indicate an error.

The frequency scale in the LabView VI is set to a range of 0 to 100,000, consistent with the first type of frequency test, this should be changed for other test types. Perform at least one frequency scan 0 for all channels.

Trim Potentiometer Adjustment

Connect the clock signal from the 53.1 MHz NIM module oscillator to the CLK IN input on the DSR board. Ensure there is at least a 7 dBm pad between the NIM oscillator output and the CLK IN input. Terminate the input of the channel of interest to 50 Ω. Choose the Trim Potentiometers test mode in the `TestDsrCTRL` VI and choose the channel of interest from drop down list. Click "Send to Hardware".

This mode is meant as a fine tune adjustment of the potentiometers for each channel. Load the LabView VI TestDsrTrim}, this may be started now or after the front end task is started. Initiate the test by clicking "START TEST".

Now look at the LabView application, and start it if you have not already done so. If the data is not visible in the graph, adjust the scale until it is visible. Now adjust the potentiometer for a the channel of interest while watching the data on the LabView application. The goal is to reduce the signal amplitude to zero. Reduce the y-scale in order to make finer adjustments as you approach the minimum. The signal may still be noisy and not be at exactly zero but there will be a minimum where the signal amplitude and oscillation are minimized. Aim for this.

Repeat this adjustment for each pair of channels, adjusting the channel potentiometers independently as above. You may need to iterate these adjustments based on the results from the power sweep procedure below.

Power Sweep Procedure

Before performing these tests it is important to trim the potentiometers for each channel as described in the previous section. In this test the HP signal generator will be the input signal source. Set up the HP signal generator to perform an amplitude sweep from +20 to -70 dBm, in steps of -3 dBm, with a point trigger value of EXT. Once the power sweep is started at the signal generator, the slot 0 will trigger all further power steps through the SWH module. The frequency of the HP signal generator should be set to 300.0 kHz. When the splitter is accounted for the input signal amplitude starts at roughly 10 dBm.

Split the input using the -9.5 dB splitter and direct the input to channels 1 and 2, terminate the third output of the splitter into 50 Ω. Connect the clock signal from the 53.1 MHz NIM module oscillator to the CLK IN input on the DSR board. Ensure there is at least a 7 dBm pad between the NIM oscillator output and the CLK IN input.

Choose the Power Sweep test mode in the `TestDsrCTRL` VI and choose the channel pair of interest from drop down list. Click “Send to Hardware”. Load the LabView VI `TestDsrPwr`, and start the VI running. At the signal generator, an amplitude sweep must be enabled and a sweep is started by pressing the “Single Sweep” button. Initiate the power sweep at the computer by clicking “START TEST”.

Now you can watch the data appear on the LabView graph. The ideal case is a horizontal line, which shows the calculated value of “position” is unaffected by the input power. If there is significant deviation from this it may be necessary to retrim the pots. If this does not solve the problem, log the error and go on.

Note: the axis in LabView gives the absolute value of the power. This scale is only loosely coupled to actual power coming from the signal generator so be wary of absolute interpretation of the x-axis values. Also, during the course of the power scan tne gain change is made at approximately -32 dBm. The graphs should not change at this point, if they do this may indicate a trimming problem.

Contacts

Brian Chase (LLRF group leader)
chase@fnal.gov
officR: TGS-xxx
tel:	x3040
pager:

Paul Joireman (Principal system programmer)
joireman@fnal.gov
officR: TGS-125
tel:	x2550

Philip Varghese (DSP programming support/pulsed mode)
varghese@fnal.gov
officR:
tel:	x4803

Index

[1] Plans for Tevatron Run IIB, Fermilab internal publication, 12/02/01, p. 120, accessible at: http://www-bdnew.fnal.gov/pbar/run2b/Documents/tdr/tdr.pdf

[2] See for examplR: Prospectus for an Electron Cooling System for the Recycler, Fermilab TM-2061, accessible at http://fnalpubs.fnal.gov/archive/1998/tm/TM-2061.pdf